A Functional Mirror Descent Perspective on Reinforcement Learning

Boris Belousov [1 2] **Maximilian Hensel** [1] **Tuan Dam** [1] **Jan Peters** [1 2 3]

Abstract

Functional mirror descent offers a unifying perspective on optimization of statistical models and provides numerous advantages for the design and analysis of learning algorithms. It brings the concepts from optimization—such as surrogate models, constraints, projections, conjugate duality, momentum—into the realm of optimization over probability distributions. So far, only a fraction of these insights have been utilized in reinforcement learning (RL), with most progress achieved in the bandit setting and in discrete MDPs, but not in the full RL setup with continuous states, observations, goals, tasks, etc. We argue for a much tighter integration of the ideas from (online) convex optimization into the design of RL algorithms for continuous MDPs by (i) showing how a number of existing approaches can be framed as approximate mirror descent on the space of probability measures and (ii) indicating yet unexplored directions uncovered by this perspective. We hope that our exposition will stimulate a wider use of advanced optimization tools in reinforcement learning and at the same time encourage the development of novel optimization approaches motivated by the RL problem.

1. Introduction

Probability functional descent (PFD) (Chu et al., 2019) has been recently proposed as a unifying meta-algorithm for optimization of probabilistic models. Remarkably, PFD is able to treat in a uniform manner such disparate algorithms as VAEs (Kingma & Welling, 2013), GANs (Goodfellow et al., 2014), and actor-critic methods (Konda & Tsitsiklis, 2000). This level of generality is enabled by framing the problems in the language of functional optimization over probability distributions. Each iteration of the PFD algorithm then consists of two steps, which are called differentiation step and descent step. In the differentiation step, a linear functional is obtained that locally approximates the true objective; in the descent step, the distribution of interest is updated to decrease the value of this approximate functional. Chu et al. (2019) furthermore describe a general technique for linearizing probability functionals based on convex duality, which underlies adversarial training methods such as Wasserstein GAN (Arjovsky et al., 2017) and dual actor-critic (Chen & Wang, 2016; Dai et al., 2018).

Although the PFD scheme can be readily applied to the expected return objective of reinforcement learning (RL), it leaves the exact implementation of the descent step unspecified. The descent step on the RL objective corresponds to policy improvement, and it is a well-established fact that some form of a conservative update is required to prevent large changes to the policy (Kakade & Langford, 2002; Peters et al., 2010; Schulman et al., 2015). In order to incorporate such conservative updates, we propose to specialize the descent step in PFD to a mirror descent (MD) step (Beck & Teboulle, 2003). Since the optimization variable is a probability distribution, the machinery of convex optimization in Banach spaces (Sridharan & Tewari, 2010) needs to be called upon. We call the resulting algorithm functional mirror descent (FMD).

The connection to mirror descent provides a link to the rich online convex optimization literature (Shalev-Shwartz et al., 2012; Bubeck et al., 2015; Orabona, 2019), where MD plays a prominent role, being a universal algorithm in the sense of achieving a (nearly) optimal regret guarantee (Srebro et al., 2011). Moreover, algorithms based on mirror descent were shown to enable Bayesian inference (BI) with provable guarantees (Dai et al., 2016). Since there is a deep connection between reinforcement learning and Bayesian inference (Rawlik et al., 2013; Levine, 2018), one may be able to transfer the insights and analysis tools from BI to RL using mirror descent as a bridge.

The rest of the paper is structured as follows. In Section 2, we provide the background on the reinforcement learning problem, paying special attention to the maximum entropy RL setting. In Section 3, we introduce probability functional descent in detail and describe how it can be applied to RL. In Section 4, we cover the basics of mirror descent and

[1] Department of Computer Science, Technical University of Darmstadt, Germany [2] German Research Center for AI (DFKI), Research Department: Systems AI for Robot Learning [3] Hessian.AI. Correspondence to: Boris Belousov <boris@robot-learning.de>.

Technical report.

present the proposed functional mirror descent algorithm. In Sections 5 and 6, we detail the FMD perspective on policy search and RL, respectively. Finally, in Sections 7– 9, we provide an overview of the frontiers, discuss related work, and conclude with an outlook towards future extensions.

2. Background on Reinforcement Learning

Reinforcement learning (Sutton & Barto, 2018) can be viewed as a problem of optimizing a controller for an unknown dynamical system through interaction with the system. To extract information from the interactions, a statistical model of the relationships between the observed variables needs to be postulated (Barber, 2012). In RL, the observed variables are the states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, and rewards $r \in \mathcal{R}$. The spaces $\mathcal{S}, \mathcal{A}, \mathcal{R}$ can have various nature; most commonly, $\mathcal{S}$ and $\mathcal{A}$ are either finite sets or finite-dimensional vector spaces, and $\mathcal{R}$ is a (bounded) subspace of $\mathbb{R}$. The set of variables can be extended with observations, goals, contexts, and other parameters that may be assumed observed; moreover, hidden variables can be introduced to represent state and action abstractions, as well as task abstractions in case of multi-task RL (Levine, 2018).

Denote the dynamics by $p(s'|s,a)$ and the reward function by $r(s,a)$. Any control law can be represented by a conditional distribution $\pi(a|s)$, called policy (Bertsekas, 2019). The performance of a given policy $\pi(a|s)$ is measured by the expected reward. Various RL settings exist: finite- or infinite-horizon, discounted or undiscounted rewards, ergodic or non-ergodic dynamical systems. For concreteness, we will consider finite-horizon undiscounted problems to follow the related works (Rawlik et al., 2013; Levine, 2018). This setting covers many cases of practical interest, including trajectory optimization, finite-horizon discounted problems (via time-dependent rewards), and infinite-horizon discounted problems (via extending the horizon and decreasing the discount factor $\gamma \in [0,1]$).

2.1. Finite-Horizon Setting

In the finite-horizon setting, state includes time, $s = [s_t, t]$ with $s_t \in \mathcal{S}'$ and $t \in \{0, 1, \ldots, T\}$. Thus, formally, the state space is $\mathcal{S} = \mathcal{S}' \times \{0, 1, \ldots, T\}$. For notational convenience, the time variable t is often separated and treated independently from s_t. We assume stationary dynamics, $p(s_{t'}, t'|s_t, t, a) = p(s_{t'}|s_t, a)\delta_{t+1}(t')$, with $\delta_{t+1}(t')$ denoting the Dirac measure concentrated at $t+1$. Marginalizing out t', we obtain $s_{t+1} \sim p(s_{t+1}|s_t, a_t)$. The reward is also time-dependent in general, $r_t = r(s_t, t, a_t)$. Discounting can be encoded as $r(s_t, t, a_t) = \gamma^t r(s_t, a_t)$ with a time-invariant $r(s_t, a_t)$. We denote $r_t(s_t, a_t) \coloneqq r(s_t, t, a_t)$.

Any policy in the finite-horizon setting can be represented by a time-dependent distribution $\pi(a|s_t, t)$. In practice, the choice of the policy representation depends on the application. For example, if we omit the dependence on s_t, we obtain a *trajectory optimization* formulation with the controller $\pi(a|t)$. Since time is discrete, $\pi(a|t)$ can be represented by a finite set of parameters, e.g., $\pi(a|t) = \delta_{a_t}(a)$ with $a_t \in \mathcal{A}, t \in \{0, 1, \ldots, T-1\}$. Alternatively, a hierarchical controller $\pi(a|t) = \mathbb{E}_{\pi(\theta)}[\pi(a|t,\theta)]$ may be considered, with a prior $\pi(\theta)$ over the global variable $\theta \in \mathbb{R}^{n_\theta}$. If we then set $\pi(a|t,\theta) = \delta_{f(\theta,t)}(a)$ with some known deterministic function $f\colon \theta, t \mapsto a$, we recover the *episodic policy search* setting. Of course, we can also directly optimize with respect to the state-conditional policy $\pi(a|s,\theta)$ parameterized by θ; this setting is known as *policy optimization* or step-based policy search. We denote $\pi_t(a|s_t) \coloneqq \pi(a|s_t, t)$.

A policy $\pi\colon s_t, t, a \mapsto \pi_t(a|s_t)$ induces a distribution $p_\pi(\tau)$ over state-action trajectories $\tau = (s_{0:T}, a_{0:T-1})$ in the form

$$p_\pi(\tau) = p(s_0) \prod_{t=0}^{T-1} p(s_{t+1}|s_t, a_t) \prod_{t=0}^{T-1} \pi_t(a_t|s_t) \tag{1}$$

where $p(s_0)$ is a fixed initial state distribution. The cumulative reward collected along a trajectory is given by

$$r(\tau) = \sum_{t=0}^{T-1} r_t(s_t, a_t). \tag{2}$$

We omit the terminal reward $r_T(s_T)$ to avoid clutter, but it can be straightforwardly added.

2.2. Maximum Entropy Reinforcement Learning

Traditionally, the objective in RL has been the expected reward $\mathbb{E}_{p_\pi(\tau)}[r(\tau)]$ (Sutton & Barto, 2018). However, this objective is ill-posed—similarly to the empirical risk minimization (ERM) objective (Guedj, 2019). Since the optimization is performed on a finite set of samples, the policy will collapse to a deterministic mapping $\pi_t(a|s_t) = \delta_{g_t(s_t)}(a)$ with some function $g\colon s_t, t \mapsto a$. This problem has been recognized early on (Williams & Peng, 1991). To make the optimization problem well-posed and obtain a stable and well-defined solution, we need to restrict the class of distributions over which the optimization is performed.

A principled and computationally convenient way to restrict the class of distributions is to add a regularization term to the objective. A natural choice is provided by the Kullback-Leibler (KL) divergence D, thoroughly studied within PAC-Bayesian theory (Guedj, 2019). In our setting, this yields

$$D(p_\pi(\tau) \| p_{\pi^0}(\tau)) = \mathbb{E}_{p_\pi(\tau)}\left[\log \frac{p_\pi(\tau)}{p_{\pi^0}(\tau)}\right] \tag{3}$$

with $p_{\pi^0}(\tau)$ denoting the trajectory distribution under a prior policy π^0. Depending on the setting, the prior policy may be available (e.g., previous plan in MPC (Wagener et al.,

2019) or learned prior in meta-RL (Amit & Meir, 2018)) or not (e.g., classical tabula rasa RL). Either way, the KL divergence factorizes over the time steps

$$D(p_\pi \| p_{\pi^0}) = \sum_{t=0}^{T-1} \mathbb{E}_{p_\pi(s_t)} \left[D(\pi_t(a|s_t) \| \pi_t^0(a|s_t)) \right] \quad (4)$$

with $p_\pi(s_t)$ denoting the marginal distribution over s_t at time step t under policy π.

If no prior policy is given, then the KL divergence in (4) reduces to the negative entropy,

$$D(p_\pi \| p_\lambda) = \sum_{t=0}^{T-1} \mathbb{E}_{p_\pi(s_t)} \left[-\mathcal{H}(\pi_t(\cdot|s_t)) \right]. \quad (5)$$

Here, $\mathcal{H}(\pi_t(\cdot|s_t)) = \mathbb{E}_{\pi_t(a|s_t)}[-\log \pi_t(a|s_t)]$, and λ refers to the Lebesgue measure over $\mathcal{A} \subseteq \mathbb{R}^{n_a}$. Adding (5) to the reward, we arrive at the entropy-regularized RL objective

$$J(\pi) = \sum_{t=0}^{T-1} \mathop{\mathbb{E}}_{p_\pi(s_t)} \left[\mathop{\mathbb{E}}_{\pi_t(a_t|s_t)} [r_t(s_t, a_t)] + \mathcal{H}(\pi_t(\cdot|s_t)) \right], \quad (6)$$

also known as the maximum entropy RL objective (MaxEnt RL) (Levine, 2018). Complementary to our derivation, which is motivated by the PAC-Bayesian learning theory, the same objective can be derived via structured variational inference (VI) on an extended graphical model with 'optimality variables' $\mathcal{O}_t \sim p(\mathcal{O}_t = 1|s_t, a_t) = \exp(r_t(s_t, a_t))$, in which case (6) plays the role of the evidence lower bound (ELBO) (Levine, 2018). The advantage of directly starting with the optimization problem (6) is its flexibility. One can seamlessly integrate other divergence functions, or turn the KL penalty into a constraint, or tune the weighting between the terms adaptively during optimization, etc. Although similar modifications can be introduced in the VI framework, they cannot be justified from the inference perspective.

3. Probability Functional Descent

Probability functional descent (PFD) (Chu et al., 2019) is a generic algorithm for gradient-based optimization of probability functionals, i.e., real-valued functions defined on the space of probability distributions. The entropy-regularized objective (6) is an example of such a functional, with a slight peculiarity that the policy is a *conditional* distribution.

3.1. Differentiation and Descent Steps

A key insight behind PFD is the observation that commonly encountered probability functionals admit local linarization. In a finite-dimensional setting, a function $f(x)$ differentiable at x_0 can be approximated by a linear function $\bar{f}(x) = f(x_0) + \langle \nabla f(x_0), x - x_0 \rangle$ around x_0. We can rewrite it as $\bar{f}(x) = \langle \nabla f(x_0), x \rangle + \text{const}$. An analog of such linearization for a probability functional $F(p)$ in the vicinity of a distribution $p_0(x)$ is given by the linear functional $\bar{F}(p) = \mathbb{E}_{p(x)}[\Psi_{p_0}(x)] + \text{const}$. Here, $\Psi_{p_0}(x)$ denotes the *influence function*, which plays the role of the gradient in the probability-functional setting.[1] Having obtained the linear approximation $\bar{F}(p)$, we subsequently perform a descent step with respect to p on this approximation.

To summarize, the PFD algorithm consists of two steps repeating in a cycle. First, in the *differentiation step*, the influence function $\Psi_{p_0}(x)$ is computed given the current distribution p_0. Second, in the *descent step*, the next distribution p is found that decreases $\mathbb{E}_{p(x)}[\Psi_{p_0}(x)]$.

3.2. Missing Links to Reinforcement Learning

A number of algorithms have been framed as instances of PFD, including optimization of generative adversarial networks (GAN) (Goodfellow et al., 2014), black-box variational inference (Ranganath et al., 2014), dual actor-critic (Dai et al., 2018), and others. Although several approaches to reinforcement learning have also been considered, the treatment of RL from the PFD perspective has been incomplete.

- MaxEnt RL objective (6) has not been addressed. Once framed as PFD, MaxEnt RL may directly benefit from the specific techniques developed for other PFD-based algorithms, e.g., influence function approximation.
- Conservative descent steps have not been incorporated into PFD. In RL, conservative updates are known to be crucial (Kakade & Langford, 2002); once embedded into PFD, they can be used in other applications.
- The differentiation step of PFD has only been identified with model-free RL. However, there are important model-based methods, such as guided policy search (GPS) (Levine & Koltun, 2013), not covered by PFD.
- Sampling has not been treated explicitly. By incorporating empirical distributions into PFD, connections between sampling and optimization over measures can be utilized (Dai et al., 2016; Wibisono, 2018).

In the remainder of the paper, we address these points, establishing a tighter connection between PFD and RL. In the next section, we recap the basics of mirror descent and introduce the functional mirror descent (FMD) algorithm. Subsequently, we demonstrate the usefulness of such extension in policy search and RL problems. Finally, we close with a discussion of the future directions uncovered by this FMD perspective.

[1] More formally, we assume $F(p)$ to be Gateaux differentiable at p_0, with the Gateaux differential admitting the integral representation $dF_{p_0}(p - p_0) = \mathbb{E}_p[\Psi_{p_0}] - \mathbb{E}_{p_0}[\Psi_{p_0}]$ (Huber, 2004).

4. Mirror Descent

Mirror descent (MD) is an extension of gradient descent to non-Euclidean spaces (Beck & Teboulle, 2003).

4.1. Mirror Descent in Finite-Dimensional Spaces

If X is a Euclidean space, then gradient descent on a differentiable convex function $f\colon X \to \mathbb{R}$ prescribes the update $x - \eta \nabla f(x)$ with some step size $\eta > 0$. Properly speaking, the gradient $\nabla f(x)$ is an element of the dual space X^*, i.e., the space of continuous linear functionals on X. We can subtract the gradient $\nabla f(x) \in X^*$ from the current point $x \in X$ only because the dual space X^* is isometric to X in the Euclidean setting. In more general settings, such operation is not possible (Bubeck et al., 2015).

Mirror descent lifts the limitation of gradient descent and enables first-order optimization in Banach spaces. MD maps a point $x \in U \subseteq X$ into the dual space X^*, performs a gradient update in X^*, then maps the resulting point back to the primal space X, and finally, performs a projection onto the subspace of interest U. Both the primal/dual mapping and the projection are based on the concept of a *mirror map*, which is closely associated with the notions of Bregman divergence and conjugate duality (Bubeck et al., 2015).

The Euclidean space X equipped with the inner product $\langle , \rangle : X \times X \to \mathbb{R}$ is a Hilbert space, i.e., a Banach space with the norm derived from the inner product. In this case, the mirror descent step can be written as

$$x_{i+1} = \operatorname*{arg\,min}_{x \in U} \langle \nabla f(x_i), x \rangle + \frac{1}{\eta_i} B_\phi(x; x_i) \tag{7}$$

where $B_\phi\colon X \times \operatorname{int} X \to \mathbb{R}$ is the Bregman divergence generated by a strictly convex, continuously differentiable on the interior of X function $\phi\colon X \to \mathbb{R}$ (Orabona, 2019). If $\phi(x) = \frac{1}{2}\|x\|_2^2$, then $B_\phi(x; x_i) = \frac{1}{2}\|x - x_i\|_2^2$ and we recover the standard (projected) gradient descent from (7). Function ϕ is called a mirror map if it is proper, its gradient $\nabla\phi(x)$ takes all possible values, and $\nabla\phi(x)$ diverges on the boundary of X (Bubeck et al., 2015). We will be chiefly interested in a special case of (7), known as *entropic mirror descent* (Beck & Teboulle, 2003), where U is the simplex and B_ϕ is the KL divergence; and more precisely, in the extension of this case to the probability functional setting. Indeed, neither the gradient, nor the inner product are defined for the functionals of the form $F(p)$ where p is a probability distribution.

4.2. Functional Mirror Descent

We now introduce the form of mirror descent applicable to probability functional optimization. Let $\mathcal{P}(\mathcal{X})$ be the space of Borel probability measures on a topological space $\mathcal{X}$. To avoid technicalities, we assume that $\mathcal{X}$ is a compact Polish space, following Chu et al. (2019). The space of probability distributions $\mathcal{P}(\mathcal{X})$ is a convex subset of the vector space of finite signed Borel measures $\mathcal{M}(\mathcal{X})$, equipped with the topology of weak convergence. The dual of $\mathcal{M}(\mathcal{X})$ is the space $\mathcal{C}(\mathcal{X})$ of continuous functions $\mathcal{X} \to \mathbb{R}$ with the uniform norm topology. With these definitions, one step of entropic mirror descent on a probability functional $F(p)$ can be written as

$$p_{i+1} = \operatorname*{arg\,min}_{p \in \mathcal{P}(\mathcal{X})} \mathbb{E}_{p(x)}[\Psi_{p_i}(x)] + \frac{1}{\eta_i} D(p \| p_i) \tag{8}$$

where $p_i \in \mathcal{P}(\mathcal{X})$ is the current distribution, $\Psi_{p_i}(x)$ is the influence function, that roughly plays the role of $\nabla F(p_i)$, $D(p \| p_i)$ is the KL divergence, and $\eta_i > 0$ is the step size.

A remarkable property of the mirror descent update (8) is that it can be rewritten as a two-step procedure (Orabona, 2019), where the first step is an unconstrained optimization problem and the second step is a projection,

$$\tilde{p}_{i+1} = \operatorname*{arg\,min}_{p \in \mathcal{M}(\mathcal{X})} \mathbb{E}_{p(x)}[\Psi_{p_i}(x)] + \frac{1}{\eta_i} D(p \| p_i), \tag{9}$$

$$p_{i+1} = \operatorname*{arg\,min}_{p \in \mathcal{P}(\mathcal{X})} D(p \| \tilde{p}_{i+1}). \tag{10}$$

This form (9)–(10) will be convenient for describing projections between different policy parameterizations in RL.

The choice of the step size sequence η_i is an important design decision. Using prior information about the properties of the objective function and the search domain, one may be able to derive optimal schedules, often in the form $\eta_i \propto 1/\sqrt{i}$. Alternatively, one may use an adaptive scheme, scaling the learning rate in inverse proportion to the square root of the sum of squared dual norms of the gradients, as done in AdaGrad (Duchi et al., 2011) and related algorithms (Orabona, 2019). In the probability functional setting, a popular approach has been to treat the divergence as a constraint with a fixed bound ε and then obtain η_i^{-1} as the optimal value of the corresponding Lagrange multiplier (Peters et al., 2010; Haarnoja et al., 2018).

In summary, the mirror descent update step (8) or, equivalently, steps (9)–(10) extend the PFD algorithm with an explicit rule for updating the search distribution $p(x)$ in the descent step. In the following, we apply this procedure to policy search and MaxEnt RL and derive a number of reinforcement learning algorithms that can be understood as employing different approximations of the influence function and different parameterizations of the search distribution.

5. Policy Search From the FMD Perspective

Before moving on to full RL, we first consider a simpler black box policy search setting (Deisenroth et al., 2013). Assume the low-level controller is given as $\pi(a|s, \theta)$ with parameters $\theta \in \Theta$. For each θ, we can run the system and observe a trajectory $\tau \sim p_\pi(\tau|\theta)$ and the reward $r(\tau)$. The distribution over trajectories $p_\pi(\tau|\theta)$ induces a distribution $p_\pi(r|\theta)$ over the rewards. Denoting the mean of the reward distribution by $\bar{r}(\theta)$, the analog of the MaxEnt objective (6) in the black box setting can be written as

$$\max_{\pi \in \mathcal{P}(\Theta)} J(\pi) = \mathbb{E}_{\pi(\theta)}[\bar{r}(\theta)] + \mathcal{H}(\pi). \tag{11}$$

Both in (6) and (11), the scaling factor between the two terms is hidden in the reward, as in (Levine, 2018).

Let $F(\pi) = -J(\pi) = -\mathbb{E}_{\pi(\theta)}[\bar{r}(\theta) - \log \pi(\theta)]$ and note that $F(\pi)$ is convex in π. Therefore, we can apply FMD (8). By inspection, we recognize $\Psi_\pi(\theta) = -\bar{r}(\theta) + \log \pi(\theta)$ as the influence function. If $\pi_i(\theta)$ is the current search distribution, then

$$\pi_{i+1}(\theta) = \operatorname*{arg\,min}_{\pi \in \mathcal{P}(\Theta)} \mathbb{E}_{\pi(\theta)}[\Psi_{\pi_i}(\theta)] + \frac{1}{\eta_i} D(\pi(\theta) \| \pi_i(\theta)). \tag{12}$$

In order to instantiate (12), we need to represent $\pi_i(\theta)$, $\pi(\theta)$, and $\Psi_{\pi_i}(\theta)$. Practical algorithms differ in the choice of the representations and in the implementation of the $\arg\min$ operator. In the following, we detail several design options.

5.1. Sampling Distribution and Policy Representations

At iteration i, a set of N query points $\theta_{1:N}$ is sampled from the current search distribution $\pi_i(\theta)$. Each point is rated by the reward $r_n \sim p_\pi(r|\theta_n)$. Thus, the dataset at iteration i is given by $\mathcal{D}_i = \{(\theta_n, r_n) \mid n = 1, 2, \ldots, N\}$. A convenient representation of $\mathcal{D}_i$ that manifestly obeys permutation invariance is the empirical distribution

$$\hat{p}_{\pi_i}(r, \theta) = \frac{1}{N} \sum_{n=1}^{N} \delta_{r_n}(r) \delta_{\theta_n}(\theta). \tag{13}$$

From $\hat{p}_{\pi_i}(r, \theta)$, one may, e.g., estimate $\Psi_{\pi_i}(\theta)$. However, apart from the information in $\mathcal{D}_i$, one may additionally leverage the knowledge of $\pi_i(\theta)$ in parametric form.

Depending on the representation of the policy, different operations may be computationally easy or hard. Most commonly, the policy is parameterized by a Gaussian distribution, which is convenient to work with because it enables straightforward sampling, cheap evaluation of the log-likelihood, and closed-form expressions for the entropy and the KL (Deisenroth et al., 2013). Nevertheless, unimodal policies may be limiting; in these cases, more general policy representations can be applied, including mixture models (Sehnke et al., 2010), energy-based models (Haarnoja et al., 2017), implicit models and normalizing flows (Tang & Agrawal, 2018). However, such representations are usually less tractable: sampling in energy-based models is expensive, entropy in normalizing flow models lacks an analytic form, log-likelihood in implicit models is not available.

5.2. Influence Function: Representation, Estimation

Along with the policy, representation of the influence function $\Psi_\pi(\theta)$ plays a crucial role. Assuming that $\log \pi(\theta)$ can be evaluated, estimation of $\Psi_\pi(\theta)$ reduces to estimation of the conditional expectation $\bar{r}(\theta) = \mathbb{E}_{p_\pi(r|\theta)}[r]$ from the data $\hat{p}_{\pi_i}(r, \theta)$. The simplest approach is to take r_n as the estimate of $\bar{r}(\theta_n)$. Formally, this corresponds to introducing a function $\hat{\bar{r}}(\theta)$ defined through the empirical distribution

$$\mathbb{E}_{\hat{p}_{\pi_i}(r,\theta)}[r] = \frac{1}{N} \sum_{n=1}^{N} r_n \delta_{\theta_n}(\theta) =: \hat{\bar{r}}(\theta) \hat{\pi}_i(\theta) \tag{14}$$

where $\hat{\pi}_i(\theta) = \frac{1}{N} \sum_{n=1}^{N} \delta_{\theta_n}(\theta)$ represents the current policy. Intuitively, $\hat{\bar{r}}(\theta)$ can be thought of as a function defined only at locations θ_n where it takes values r_n.

Although straightforward to implement, the empirical representation (14) may be suboptimal. If prior knowledge on the properties of $\bar{r}(\theta)$ is available, e.g., if $\bar{r}(\theta)$ is known to be continuous, then one can fit a more suitable model. The problem of estimating a function $\theta \mapsto \bar{r}(\theta)$ from data is a core machine learning task and a variety of algorithms exist. A popular approach is to fit a quadratic surrogate model, which works well when the policy is Gaussian, because then the $\arg\min$ operator can be implemented in closed form (Abdolmaleki et al., 2015). However, with growing dimensionality, fitting a quadratic model becomes expensive.

5.3. Distribution Propagation Through Optimization

The optimization step (12) can be seen as defining certain dynamics in the distribution space: it takes $\pi_i(\theta)$ as input, passes it through a transformation defined by the influence function, and outputs $\pi_{i+1}(\theta)$. Therefore, to obtain $\pi_{i+1}(\theta)$, we essentially need to simulate these propagation dynamics.

Some options: (i) sample particles from $\pi_i(\theta)$, pass them through the dynamics (12), fit a parametric $\pi_{i+1}(\theta)$ at the output (Peters et al., 2010); (ii) approximate terms in (12) in a way that enables closed-form propagation (Abdolmaleki et al., 2015); (iii) directly optimize (12) with respect to a parameterized policy (Schulman et al., 2015); (iv) use purely particle-based representation for both the input and the output distributions (Liu et al., 2017).

There are certainly more ways to simulate the dynamics (12), and the methods mentioned here should merely serve as illustration. In the next subsection, we will take a closer look at scheme (i) to highlight the idea of projections between policy representations based on the decomposition (9)–(10).

5.4. Projections Between Policy Representations

The search distribution $\pi_i(\theta)$ is typically given in parametric form, e.g., as a Gaussian. However, interaction with the system is always sample-based. Therefore, no matter what propagation method is used, projection of $\pi_i(\theta)$ down to the empirical distribution $\hat{\pi}_i(\theta)$ is unavoidable. Note that $\hat{\pi}_i(\theta)$ is in fact parameterized by two sets of parameters: weights of the particles and their locations. Therefore, we can change $\hat{\pi}_i(\theta)$ by either reweighting the particles or by moving them. Here, we analyze the reweighting scheme.

5.4.1. Propagation of Particles

The KL divergence in (12) is only well-defined if $\pi(\theta)$ is absolutely continuous with respect to $\pi_i(\theta)$. If we substitute the empirical distribution $\hat{\pi}_i(\theta)$ for $\pi_i(\theta)$, then $\pi(\theta)$ will concentrate on its support and take the parametric form

$$\hat{\pi}(\theta|w) = \frac{1}{N}\sum_{n=1}^{N} w_n \delta_{\theta_n}(\theta) \tag{15}$$

with $w \in \mathcal{W}(N) = \{w_{1:N} \mid w_n > 0, \sum_{n=1}^{N} w_n = N\}$ the adjustable weights of the particles. Upon substitution of $\hat{\pi}_i(\theta)$ and $\hat{\pi}(\theta|w)$ into (12), the problem reduces to a finite-dimensional optimization problem

$$\min_{w\in\mathcal{W}(N)} \frac{1}{N}\sum_{n=1}^{N} w_n \left(\hat{\Psi}_{\pi_i}(\theta_n) + \frac{1}{\eta_i}\log w_n\right) \tag{16}$$

where $\hat{\Psi}_{\pi_i}(\theta_n) = -r_n + \log \pi_i(\theta_n)$ denotes the empirical estimate of the influence function. The weights w can be found in closed form

$$w_n = \frac{N}{Z}\mathrm{e}^{-\eta_i \hat{\Psi}_{\pi_i}(\theta_n)}, \quad Z = \sum_{n=1}^{N} \mathrm{e}^{-\eta_i \hat{\Psi}_{\pi_i}(\theta_n)}, \tag{17}$$

which renders (15) with weights (17) as the optimal particle-based representation $\hat{\pi}_{i+1}(\theta)$ of $\pi_{i+1}(\theta)$.

5.4.2. Projection Onto Parametric Form

The next step is to project $\hat{\pi}_{i+1}(\theta)$ onto a parametric distribution $\pi_{i+1}(\theta)$. Denoting by $\hat{w}(\theta)$ a function that returns w_n when evaluated at θ_n, analogously to $\hat{r}(\theta)$ in (14), we obtain $\hat{\pi}_{i+1}(\theta) = \hat{w}(\theta)\hat{\pi}_i(\theta)$. According to the two-step reformulation of mirror descent (9)–(10), we now need to minimize $D(\pi_{i+1}(\theta)\|\hat{\pi}_{i+1}(\theta))$ in order to project the particle-based representation $\hat{\pi}_{i+1}(\theta)$ onto the parameterized representation $\pi_{i+1}(\theta)$, e.g., a Gaussian. In general, such projection is problematic, because $\hat{\pi}_{i+1}(\theta)$ is concentrated on a finite support. However, assuming that $\pi_i(\theta)$ is known in parametric form, we can leverage this knowledge and perform the projection as $\min_{\pi_{i+1}(\theta)} D(\pi_{i+1}(\theta)\|\hat{w}(\theta)\pi_i(\theta))$, which results in the minimization objective

$$\min_{\pi_{i+1}(\theta)} \mathbb{E}_{\hat{\pi}_i(\theta)}\left[\frac{\pi_{i+1}(\theta)}{\pi_i(\theta)}\left(-\log \hat{w}(\theta)\right)\right] + D(\pi_{i+1}\|\pi_i). \tag{18}$$

This objective can be optimized, e.g., via gradient descent on the parameters of $\pi_{i+1}(\theta)$. In agreement with the general equivalence between the one-step (8) and two-step (9)–(10) representations of mirror descent, Objective (18), at which we arrived by first propagating the particles (15)–(17) and then fitting a distribution, is equivalent to the objective (12), had we directly parameterized $\pi(\theta)$ and searched in the space of parameterized distributions.

The key observation in this subsection is that one can utilize the mirror descent decomposition (9)–(10) to transition between policy representations. Originally, $\mathcal{P}(\mathcal{X})$ must be a convex subset of the vector space $\mathcal{M}(\mathcal{X})$. However, once we start working with parameterized distributions, such as Gaussians, an irreducible approximation error gets introduced, and exact projection is no longer possible. Nevertheless, in practice, it may still be advantageous to utilize the decomposition (9)–(10) if computations in some representation can be performed more efficiently, e.g., in closed form. A notable application of the projections is the family of guided policy search (GPS) algorithms (Montgomery & Levine, 2016), where the policy is first obtained in the form of a locally-linear control law and subsequently projected onto a neural network representation. We will consider this and other examples in the next section.

6. MaxEnt RL From the FMD Perspective

Now we turn our attention to optimizing the MaxEnt RL objective (6) with functional mirror descent (8). Conceptually, the procedure is the same as in the black box setting: we first need to represent the objective by a linear functional, i.e., find the influence function, and then perform a mirror descent step on the policy, potentially followed by a projection. The influence function will be related to the value function, and therefore the differentiation step of FMD will roughly correspond to policy evaluation. The descent step will play the role of policy improvement; while the projection step will act as an imitation learning subroutine. In the following, we explain these connections in detail and highlight various approximations that have been considered in the literature.

6.1. Influence Function

We denote $F(\pi) = -J(\pi)$ and repeat the objective here for convenience in a slightly different form

$$F(\pi) = -\sum_{t=0}^{T-1} \mathop{\mathbb{E}}_{p_\pi(s_t)\pi_t(a_t|s_t)}\left[r_t(s_t,a_t) - \log \pi_t(a_t|s_t)\right]. \tag{19}$$

Objective (19) is similar to the black box objective (11), but with the important difference that the distribution $\pi_t(a_t|s_t)$ in (19) not only affects the current time step but also all future time steps $t < t' \le T$ through the marginals $p_\pi(s_{t'})$. Due to the recursive structure of the dynamics, the influence of the distribution $\pi_t(a_t|s_t)$ at time t on the future rewards can be summarized in the MaxEnt Q-function $Q_t^\pi(s_t,a_t)$, defined in such a way that the influence function $\Psi_{\pi_t}(s_t, a_t)$ with respect to $\pi_t(a_t|s_t)$ is given by

$$\Psi_{\pi_t}(s_t, a_t) = -p_\pi(s_t)\left(Q_t^\pi(s_t, a_t) - \log \pi_t(a_t|s_t)\right). \tag{20}$$

Note that $Q_t^\pi(s_t, a_t)$ not only includes the expected sum of future rewards but also the sum of future entropies; that is why we call it MaxEnt Q-function.

6.2. Mirror Descent Step

Once the influence function (20) corresponding to the distribution over actions $\pi_t^i(a_t|s_t)$ at current iteration i is given, the next distribution $\pi_t^{i+1}(a_t|s_t)$ can be found by solving

$$\min_{\pi_t(a_t|s_t)} \mathop{\mathbb{E}}_{p_\pi^i(s_t)}\left[\mathop{\mathbb{E}}_{\pi_t(a_t|s_t)}\left[-Q_t^i(s_t, a_t) + \log \pi_t^i(a_t|s_t)\right] + \frac{1}{\eta_t^i}\, D(\pi_t(\cdot|s_t)\|\pi_t^i(\cdot|s_t))\right] \tag{21}$$

where $Q^i := Q^{\pi^i}$, $p_\pi^i := p_{\pi^i}$, and $\eta_t^i > 0$. This objective is almost the same as (12) apart from one additional expectation over states. Therefore, the same considerations apply regarding how to update π_t^i to π_t^{i+1}, i.e., in closed form, using particles, by optimizing parameters of π_t directly, etc. A similar objective to (21) was considered by Akrour et al. (2016), albeit with trust region constraints on the entropy and the KL divergence rather than penalties as here.

6.3. Approximations

The key practical considerations regarding the implementation of (21) echo those in the black box setting: (i) policy representation, (ii) influence function estimation, (iii) optimization procedure. These points are interrelated, as a certain influence function representation may require a matching policy representation, which in turn may determine the optimization procedure. Therefore, all three points need to be tackled jointly. Below we highlight a few approaches, but in general the design space is quite large.

Estimation of the influence function (20) requires the state marginals $p_\pi(s_t)$, the MaxEnt Q-function $Q_t^\pi(s_t, a_t)$, and the log term $\log \pi_t(a_t|s_t)$. The log term is usually available, except for non-invertible implicit models, in which case it needs to be additionally estimated. The state marginal can either be represented by an explicit distribution, such as a Gaussian (Deisenroth & Rasmussen, 2011; Levine & Koltun, 2013), or through samples (Levine, 2018). The MaxEnt Q-function $Q_t^\pi(s_t, a_t)$ may be estimated via Monte Carlo rollouts (Akrour et al., 2016) or represented as a parametric function (Levine & Koltun, 2013) and fitted by minimizing a squared value prediction error (Levine, 2018) or a path consistency objective (Nachum et al., 2017).

The policy is commonly represented as a Gaussian. In MaxEnt RL, there is a particular interplay between the policy and the MaxEnt Q-function that can be utilized: namely, at optimum, $\pi_t^\star(a_t|s_t) \propto \exp(Q_t^\star(s_t, a_t))$, and therefore the policy may be implicitly represented as an energy-based model—an approach known as soft Q-learning (Haarnoja et al., 2017). However, sampling and normalization become expensive and require approximations. Soft actor-critic (SAC) (Haarnoja et al., 2018) avoids such problems by projecting the energy-based policy onto a Gaussian.

6.4. Policy Projections

There is often a trade-off between exactly solving an approximate problem and approximately solving an exact problem. By utilizing mirror descent, one can combine the advantages of both approaches. One method that enables such synthesis is mirror descent guided policy search (MDGPS) (Montgomery & Levine, 2016).

The idea of MDGPS is to fit a linear-quadratic model to the observed trajectories under the current policy, then find the optimal linear controller for the linearized system in the vicinity of the current controller, and finally, project the linear controller onto a global non-linear representation, e.g., a neural network policy. This scheme precisely follows the two-step procedure (9)–(10) of mirror descent. First, the exact influence function for the approximate linearized model is computed. Then the minimization step (9) is performed exactly but with respect to a linearized version of the global controller. Finally, the newly found linear controller is projected onto the global nonlinear policy that can afterwards approximately control the exact nonlinear system.

The projection step (10) essentially encompasses an imitation learning problem. Therefore, imitation learning is naturally covered by the FMD framework as a projection between different policy representations. This interpretation can be utilized for learning policies in unconventional forms, e.g., represented by programs (Verma et al., 2019).

7. Online Learning and Frontiers

The presented mirror descent perspective provides a bridge between optimization and reinforcement learning. Based on this connection, further ideas from optimization can be brought into RL. Mirror descent can be employed in model predictive control (MPC) (Wagener et al., 2019), to utilize the similarity between trajectory optimization problems at subsequent iterations in the form of dynamic mirror descent (DMD) (Hall & Willett, 2013). Predictive models in general can be incorporated into policy optimization procedures (Cheng et al., 2019) based on optimistic mirror descent (OMD) (Rakhlin & Sridharan, 2013). Trajectory optimization can be embedded into the policy learning loop via mirror descent guided policy search (MDGPS) (Montgomery & Levine, 2016). Natural evolution strategies can be analyzed using the tools developed for mirror descent (Ye & Zhang, 2019). Momentum and acceleration can be brought into policy search (Miyashita et al., 2018). By exploiting the link to optimization over probability distributions, particle-based approaches can be introduced into RL (Liu et al., 2017). Conjugate duality can be utilized to derive provably convergent algorithms (Dai et al., 2018) and perform offline policy evaluation and improvement (Nachum & Dai, 2020).

8. Related Work

Mirror descent (Nemirovski & Yudin, 1983; Beck & Teboulle, 2003) is a general optimization algorithm and it can be applied in various settings. Depending on whether exact or stochastic gradients are available, one discerns between deterministic and stochastic mirror descent (Nemirovski et al., 2009). If the loss function at each iteration is allowed to change, then the algorithm is called online mirror descent (Shalev-Shwartz et al., 2012).

Stochastic mirror descent applied directly to the policy parameters is called mirror policy optimization (MPO) (Yang & Zhang, 2019). Mirror descent on the parameters of the search distribution in the black box setting is known as mirror natural evolution strategy (MiNES) (Ye & Zhang, 2019). Mirror descent has also been extensively studied in applications to TD-learning (Mahadevan et al., 2014). In this paper, we apply mirror descent on the space of probability distributions. This results in the optimization problem (8) that can be subsequently solved by various methods, depending on the parameterizations of the distributions and functions.

A number of papers have studied entropy-regularized MDPs. Neu et al. (2017) presented a unifying view of entropy- and KL-regularized ergodic MDPs with the average reward criterion. Geist et al. (2019) introduced regularized modified policy iteration (reg-MPI) and mirror descent modified policy iteration (MD-MPI) in tabular MDPs with the infinite-horizon discounted reward criterion. Vieillard et al. (2019a) presented connections to classical optimization algorithms, including links between Frank-Wolfe and policy iteration, and Politex (Abbasi-Yadkori et al., 2019) and dual averaging (Nemirovski et al., 2009). Belousov & Peters (2019) formulated a framework based on f-divergence regularization in the average reward setting and Lee et al. (2019) developed a unified view based on Tsallis entropy regularization for the infinite-horizon discounted reward setting.

In contrast to these prior works, we consider finite-horizon problems and aim for a theory applicable to continuous state-action spaces. By extending the framework of probability functional descent (Chu et al., 2019) with mirror descent in Banach spaces (Sridharan & Tewari, 2010), we are able to cover practical algorithms, such as soft actor-critic (Haarnoja et al., 2018) and soft Q-learning (Haarnoja et al., 2017) along with many others, as parametric schemes approximating the exact mirror descent steps (20)–(21).

The perspective taken in this paper is closely related to the control-as-inference view (Rawlik et al., 2013). We refer the reader to (Levine, 2018) for a recent survey with exhaustive links to prior work. Although we derive the MaxEnt RL objective (6) from a regularization rather than inference perspective, this may be considered a minor detail from the practical point of view. More importantly, we apply mirror descent (21) to this objective—something which is not possible within the control-as-inference framework but which is very natural once (6) is viewed as an optimization objective on the space of probability distributions.

9. Discussion and Conclusion

Tremendous progress has been made in online convex optimization in the last decade (McMahan, 2017). Mirror descent has been at the heart of these developments thanks to its universality (Srebro et al., 2011). Although attempts have been made to establish connections between reinforcement learning and online optimization and bring the power of mirror descent into the realm of MDPs, prior approaches either applied mirror descent directly to parameterized policies or considered tabular settings. In this paper, we describe a meta-algorithm called functional mirror descent (FMD) that is applicable to problems with continuous state-action spaces. The FMD algorithm combines insights from probability functional descent (PFD) (Chu et al., 2019) and mirror descent in Banach spaces (Sridharan & Tewari, 2010).

We view the RL objective (6) as a probability functional and apply the FMD algorithm (8) to it. By clearly separating the algorithmic steps of (i) linearizing the functional (i.e., finding the influence function), (ii) performing the mirror descent step (9) in a convenient form (e.g., linear or non-parametric), and (iii) projecting the found policy onto a global representation (10), we are able to treat a variety of RL algorithms in a uniform manner and classify them on the basis of how they parameterize each step and what optimization procedure they employ.

The mirror descent perspective unlocks several avenues for developing improved RL methods. In one direction, better approximations of the MD step (8) can be sought. More versatile representations, such as energy-based models (Dai et al., 2019), normalizing flows (Tang & Agrawal, 2018), or interacting particle systems (Liu et al., 2019) may provide a powerful boost by leveraging the link between sampling and optimization in the space of measures (Wibisono, 2018). On the other hand, more tools from optimization can be brought into RL, such as Fenchel-Rockafellar duality (Nachum & Dai, 2020), momentum Vieillard et al. (2019b) and acceleration Miyashita et al. (2018).

References

Abbasi-Yadkori, Y., Bartlett, P., Bhatia, K., Lazic, N., Szepesvari, C., and Weisz, G. Politex: Regret bounds for policy iteration using expert prediction. In *International Conference on Machine Learning*, pp. 3692–3702, 2019.

Abdolmaleki, A., Lioutikov, R., Peters, J. R., Lau, N., Reis, L. P., and Neumann, G. Model-based relative entropy stochastic search. In *Advances in Neural Information Processing Systems*, pp. 3537–3545, 2015.

Akrour, R., Neumann, G., Abdulsamad, H., and Abdolmaleki, A. Model-free trajectory optimization for reinforcement learning. In *International Conference on Machine Learning*, pp. 2961–2970, 2016.

Amit, R. and Meir, R. Meta-learning by adjusting priors based on extended pac-bayes theory. In *International Conference on Machine Learning*, pp. 205–214, 2018.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pp. 214–223, 2017.

Barber, D. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

Beck, A. and Teboulle, M. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Belousov, B. and Peters, J. Entropic regularization of markov decision processes. *Entropy*, 21(7):674, 2019.

Bertsekas, D. *Reinforcement learning and optimal control*. Athena Scientific, 2019.

Bubeck, S. et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Chen, Y. and Wang, M. Stochastic primal-dual methods and sample complexity of reinforcement learning, 2016.

Cheng, C.-A., Yan, X., Ratliff, N., and Boots, B. Predictor-corrector policy optimization. In *International Conference on Machine Learning (ICML)*, 2019.

Chu, C., Blanchet, J., and Glynn, P. Probability functional descent: A unifying perspective on gans, variational inference, and reinforcement learning. In *International Conference on Machine Learning*, pp. 1213–1222, 2019.

Dai, B., He, N., Dai, H., and Song, L. Provable bayesian inference via particle mirror descent. In *Artificial Intelligence and Statistics*, pp. 985–994, 2016.

Dai, B., Shaw, A., Li, L., Xiao, L., He, N., Liu, Z., Chen, J., and Song, L. Sbeed: Convergent reinforcement learning with nonlinear function approximation. In *International Conference on Machine Learning*, pp. 1125–1134, 2018.

Dai, B., Liu, Z., Dai, H., He, N., Gretton, A., Song, L., and Schuurmans, D. Exponential family estimation via adversarial dynamics embedding. In *Advances in Neural Information Processing Systems*, pp. 10977–10988, 2019.

Deisenroth, M. and Rasmussen, C. E. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pp. 465–472, 2011.

Deisenroth, M. P., Neumann, G., Peters, J., et al. A survey on policy search for robotics. *Foundations and Trends® in Robotics*, 2(1–2):1–142, 2013.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12:2121–2159, 2011.

Geist, M., Scherrer, B., and Pietquin, O. A theory of regularized markov decision processes. In *International Conference on Machine Learning*, pp. 2160–2169, 2019.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Guedj, B. A primer on pac-bayesian learning. *arXiv preprint arXiv:1901.05353*, 2019.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1352–1361, 2017.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Hall, E. C. and Willett, R. M. Dynamical models and tracking regret in online convex programming. In *International Conference on Machine Learning*, pp. I–579, 2013.

Huber, P. J. *Robust statistics*, volume 523. John Wiley & Sons, 2004.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, pp. 267–274, 2002.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Konda, V. R. and Tsitsiklis, J. N. Actor-critic algorithms. In *Advances in neural information processing systems*, pp. 1008–1014, 2000.

Lee, K., Kim, S., Lim, S., Choi, S., and Oh, S. Tsallis reinforcement learning: A unified framework for maximum entropy reinforcement learning. *arXiv preprint arXiv:1902.00137*, 2019.

Levine, S. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Levine, S. and Koltun, V. Guided policy search. In *International Conference on Machine Learning*, pp. 1–9, 2013.

Liu, C., Zhuo, J., Cheng, P., Zhang, R., and Zhu, J. Understanding and accelerating particle-based variational inference. In *International Conference on Machine Learning*, pp. 4082–4092, 2019.

Liu, Y., Ramachandran, P., Liu, Q., and Peng, J. Stein variational policy gradient. In *33rd Conference on Uncertainty in Artificial Intelligence, UAI 2017*, 2017.

Mahadevan, S., Liu, B., Thomas, P., Dabney, W., Giguere, S., Jacek, N., Gemp, I., and Liu, J. Proximal reinforcement learning: A new theory of sequential decision making in primal-dual spaces. *arXiv preprint arXiv:1405.6757*, 2014.

McMahan, H. B. A survey of algorithms and analysis for adaptive online learning. *The Journal of Machine Learning Research*, 18(1):3117–3166, 2017.

Miyashita, M., Yano, S., and Kondo, T. Mirror descent search and its acceleration. *Robotics and Autonomous Systems*, 106:107–116, 2018.

Montgomery, W. H. and Levine, S. Guided policy search via approximate mirror descent. In *Advances in Neural Information Processing Systems*, pp. 4008–4016, 2016.

Nachum, O. and Dai, B. Reinforcement learning via fenchel-rockafellar duality. *arXiv preprint arXiv:2001.01866*, 2020.

Nachum, O., Norouzi, M., Xu, K., and Schuurmans, D. Bridging the gap between value and policy based reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 2775–2785, 2017.

Nemirovski, A. and Yudin, D. *Problem complexity and method efficiency in optimization*. Wiley, New York, 1983.

Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

Neu, G., Jonsson, A., and Gómez, V. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

Orabona, F. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.

Peters, J., Mulling, K., and Altun, Y. Relative entropy policy search. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

Rakhlin, A. and Sridharan, K. Online learning with predictable sequences. In *Conference on Learning Theory*, pp. 993–1019, 2013.

Ranganath, R., Gerrish, S., and Blei, D. Black box variational inference. In *Artificial Intelligence and Statistics*, pp. 814–822, 2014.

Rawlik, K., Toussaint, M., and Vijayakumar, S. On stochastic optimal control and reinforcement learning by approximate inference. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897, 2015.

Sehnke, F., Graves, A., Osendorfer, C., and Schmidhuber, J. Multimodal parameter-exploring policy gradients. In *2010 Ninth International Conference on Machine Learning and Applications*, pp. 113–118. IEEE, 2010.

Shalev-Shwartz, S. et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Srebro, N., Sridharan, K., and Tewari, A. On the universality of online mirror descent. In *Advances in neural information processing systems*, pp. 2645–2653, 2011.

Sridharan, K. and Tewari, A. Convex games in banach spaces. In *COLT*, pp. 1–13. Citeseer, 2010.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Tang, Y. and Agrawal, S. Implicit policy for reinforcement learning. *arXiv preprint arXiv:1806.06798*, 2018.

Verma, A., Le, H., Yue, Y., and Chaudhuri, S. Imitation-projected programmatic reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 15726–15737, 2019.

Vieillard, N., Pietquin, O., and Geist, M. On connections between constrained optimization and reinforcement learning. *arXiv preprint arXiv:1910.08476*, 2019a.

Vieillard, N., Scherrer, B., Pietquin, O., and Geist, M. Momentum in reinforcement learning. *arXiv preprint arXiv:1910.09322*, 2019b.

Wagener, N., Cheng, C.-A., Sacks, J., and Boots, B. An online learning approach to model predictive control. In *Proceedings of Robotics: Science and Systems (RSS)*, 2019.

Wibisono, A. Sampling as optimization in the space of measures: The langevin dynamics as a composite optimization problem. In *Conference On Learning Theory*, pp. 2093–3027, 2018.

Williams, R. J. and Peng, J. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268, 1991.

Yang, L. and Zhang, Y. Policy optimization with stochastic mirror descent. *arXiv preprint arXiv:1906.10462*, 2019.

Ye, H. and Zhang, T. Mirror natural evolution strategies. *arXiv preprint arXiv:1910.11490*, 2019.